MCDI
MAESTRÍA EN CIENCIA DE DATOS E INFORMACIÓN

INFOTEC CENTRO DE INVESTIGACIÓN E INNOVACIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

DIRECCIÓN ADJUNTA DE INNOVACIÓN Y CONOCIMIENTO
GERENCIA DE CAPITAL HUMANO
POSGRADOS

"Predicción de Fallos en redes Wi-Fi mediante aprendizaje computacional"

Tesis
Que para obtener el grado de MAESTRO EN CIENCIA DE DATOS E INFORMACIÓN

Presenta:

Hugo Hernández Aceves

Asesor:

Mario Graff Guerrero

Ciudad de México, mayo, 2025.

CONACYT

INFOTEC POSGRADOS

Autorización de impresión

Transformación Digital
Agencia de Transformación Digital y Telecomunicaciones

INFOTEC

Declaración de Autorización de Depósito de Recursos de Información en el Repositorio Institucional de INFOTEC

Datos de la persona DEPOSITARIA	Datos del RECURSO DE INFORMACIÓN
Nombre(s): Hugo Apellidos: Hernández Aceves **Señale con una X el dato correcto:** Autor/Autora: Único(a) X Coautoría ___ Clasificación: Investigador(a) ___ Académico(a) ___ Estudiante de Posgrado o TSU X Tecnólogo(a) ___ Otro (describa) _____	Título: Predicción de fallos en redes Wi-Fi mediante aprendizaje computacional. **Señale con una X el dato correcto:** ¿El recurso de información ha sido publicado previamente? Si ___ No X ¿El recurso de información ha sido sometido a un proceso de evaluación para ser publicado? Si ___ No ___ Clasificación: ☒ Literaria ☐ Música ☐ Audiovisual ☐ Software ☐ Otro(especificar):

Yo Hugo Hernández Aceves, en mi carácter de persona autora, declaro:

I. Que soy la persona autora o una de las personas autoras, de una creación original y primigenia que no invade derechos de terceros y/o de todos los recursos de información, académica, científica, tecnológica y de innovación original, la cual será depositada en el Repositorio Institucional de Acceso Abierto a Recurso de Información Académica, Científica, Tecnológica y de Innovación de INFOTEC, de calidad e interés social y cultural, en adelante "Repositorio Institucional", al que se hace referencia en el Decreto por el que se reforman y adicionan diversas disposiciones de la Ley de Ciencia y Tecnología, de la Ley General de Educación y de la Ley Orgánica del Consejo Nacional de Ciencia y Tecnología, publicada el 20 de mayo de 2014 en el Diario Oficial de la Federación; así como, en los Lineamientos Generales para el Repositorio Nacional y los Repositorios Instruccionales con fecha del 20 de noviembre de 2014 y los Lineamientos Técnicos para el Repositorio Nacional y Repositorios Institucionales con fecha del 26 de noviembre de 2015.

II. Que entiendo que el Repositorio Institucional es una plataforma digital cuya coordinación y modelos de operación serán emitidos por INFOTEC, siguiendo estándares internacionales para el almacenamiento, mantenimiento, preservación y diseminación de la información académica, científica, tecnológica y de innovación, la cual, se deriva de las investigaciones, productos educativos y académicos, en adelante "Recursos de información"

2025 Año de La Mujer Indígena

III. Que manifiesto la intención de que mi obra sea divulgada a través del "Repositorio Institucional", y por tal motivo otorgo mi autorización de forma expresa respecto de los "Recursos de Información" a depositar o, en su caso, cuento con la autorización expresa y comprobable para realizar el proceso de depósito del archivo en el Repositorio Nacional conforme a lo establecido en los Lineamientos Técnicos para el Repositorio Nacional y los Repositorios Institucionales.

IV. Que me he asegurado de efectuar los trámites y procesos correspondientes para la obtención de los derechos de Propiedad Intelectual de mi "Recurso de Información" a fin de obtener o reservarme los derechos aplicables necesarios para realizar el depósito legal del archivo en el "Repositorio Institucional" y que esta acción no viola de ninguna manera las disposiciones aplicables y vigentes en materia de protección de la propiedad industrial y derechos de autor, seguridad nacional, y demás regulaciones aplicables.

V. Que el "Recurso de Información" depositado no ha sido clasificado como información confidencial o reservada por alguna autoridad competente, en términos de lo dispuesto por la Ley General de Transparencia y Acceso a la Información Pública, Ley Federal de Transparencia y Acceso a la Información Pública y demás normatividad aplicable.

VI. Que las consideraciones, características y atributos del "Recurso de Información" publicaciones científicas, productos del desarrollo tecnológico y la innovación y de los datos de las investigaciones que se depositarán en el "Repositorio Institucional" estará de conformidad con lo establecido en los Artículos 1 al 4 de los Lineamientos Técnicos para el Repositorio Nacional y los Repositorios Institucionales según corresponda.

VII. Que las gráficas, tablas y/o figuras; así como, cualquier cita textual o no textual contenidas en el "Recurso de Información" científica, tecnológica y de innovación depositados están debidamente referenciadas en un estilo conocido y apropiado para la disciplina, la ciencia o el campo de conocimiento.

VIII. Que es de mi conocimiento que el INFOTEC se deslinda de toda y cualquier responsabilidad adjudicable por el indebido uso de los recursos depositados.

IX. Que reconozco que la función principal del "Repositorio Institucional" es el acopio, preservación, gestión y acceso electrónico a la información y contenidos de calidad, incluyendo aquellos de interés social y cultural que se producen en México con recursos públicos, de acuerdo con el Capítulo III, artículo 12, de los Lineamientos Generales para el Repositorio Nacional y los Repositorios Institucionales.

Habiendo manifestado lo anterior, otorgo de manera expresa la presente autorización y depósito, consistente en una licencia de carácter no exclusivo al operador del "Repositorio Institucional", para la reproducción, divulgación, distribución, comunicación y comunicación pública del "Recurso de Información", de conformidad con la licencia modelo Creative Commons, entendiéndose por ésta al conjunto de licencias públicas de libre difusión que complementan los Derechos de Autor y fomentan la colaboración; así como, de la distribución y uso de los materiales creativos, reconociendo en todo momento la autoría sobre los "Recursos de Información", los cuales no serán utilizados para fines comerciales o para generar obras derivadas de la información depositada.

Como persona depositaria, me comprometo a informar oportunamente al "Repositorio Institucional" o sus administradores, cualquier circunstancia, concesión de licencia o transmisión, o la celebración de cualquier acto jurídico que limite, restringa o impida la reproducción, divulgación, distribución, comunicación y comunicación pública de los "Recursos de Información".

Hugo Hernández Aceves

Nombre y firma de la persona depositaria

Agradecimientos

A las mujeres que me forjaron; Anita, Liza, Gloria y Paty.

A la voluntad que siempre me grita «avanza».

Índice general

Índice de figuras

Índice de tablas

Abreviaturas y acrónimos

AP: Access Point

IA: Inteligencia Artificial

SVM: Support Vector Machine (Máquina de Soporte Vectorial)

SLA: Service Level Agreement (Acuerdo de Nivel de Servicio)

TI: Tecnologías de la Información

Wi-Fi: Wireless Fidelity

Glosario

A

- **Access Point (AP):** Dispositivo que permite la conexión inalámbrica de dispositivos a una red cableada, extendiendo la cobertura de la señal Wi-Fi.
- **Aggregate Fragment Throughput:** Medida del rendimiento total de los fragmentos de datos transmitidos en una red.
- **Algoritmo de Clasificación:** Método de aprendizaje automático utilizado para categorizar datos en diferentes clases o categorías predefinidas basándose en patrones aprendidos de datos históricos.
- **Algoritmo de Clusterización:** Técnica de agrupamiento que organiza un conjunto de datos en grupos (clusters) donde los objetos dentro de cada grupo son más similares entre sí que a los de otros grupos.
- **Availability:** Capacidad de un sistema o servicio para estar operativo y accesible durante un periodo de tiempo determinado.

C

- **Ciencia de Datos:** Campo interdisciplinario que utiliza técnicas estadísticas, algoritmos de aprendizaje automático y sistemas de inteligencia artificial para extraer conocimientos y patrones de grandes volúmenes de datos.
- **Conectividad:** Capacidad de los dispositivos para comunicarse entre sí dentro de una red, facilitando el intercambio de datos.
- **CPU Usage:** Porcentaje de uso del procesador en un sistema.

D

- **Dropped Throughput:** Cantidad de datos perdidos en la transmisión de una red debido a congestión u otros problemas.
- **Dropped Throughput (packets per second):** Número de paquetes de datos perdidos por segundo en una red.

I

- **IO Memory Usage:** Cantidad de memoria utilizada para operaciones de entrada y salida en un sistema.
- **Inbound Discards:** Paquetes de datos descartados en el tráfico entrante de una red.
- **Inbound Errors:** Errores detectados en los paquetes de datos recibidos en una red.
- **Inbound Traffic:** Cantidad de datos entrantes en una red.
- **Inteligencia Artificial (IA):** Conjunto de técnicas y algoritmos que permiten a los sistemas informáticos realizar tareas que requieren razonamiento, como la predicción de fallos en redes mediante aprendizaje automático.

L

- **Latency:** Retardo en la transmisión de datos dentro de una red.
- **Load Average:** Promedio de carga del sistema medido en un intervalo de tiempo.

M

- **Máquina de Soporte Vectorial (SVM):** Algoritmo de aprendizaje supervisado utilizado para clasificación y regresión, que busca encontrar el hiperplano que mejor separa las diferentes clases en un conjunto de datos.
- **Memory Usage:** Cantidad de memoria utilizada por un sistema en un momento determinado.
- **Modelo Predictivo:** Construcción matemática o computacional diseñada para estimar la probabilidad de ocurrencia de eventos futuros o valores desconocidos, a partir de datos históricos y actuales. Estos modelos se fundamentan en técnicas estadísticas, de aprendizaje automático o inteligencia artificial, que permiten identificar relaciones complejas, patrones y dependencias entre variables. Su objetivo principal es generalizar

adecuadamente sobre nuevos datos, minimizando el error de predicción y maximizando su capacidad de inferencia sobre contextos no observados.

N

- **No Buffer Dropped Throughput (packets per second):** Número de paquetes descartados por segundo debido a la falta de espacio en el buffer.

O

- **Operational Status:** Estado operativo actual de un sistema o dispositivo en una red.
- **Outbound Discards:** Paquetes de datos descartados en el tráfico saliente de una red.
- **Outbound Errors:** Errores detectados en los paquetes de datos enviados desde un sistema.
- **Outbound Traffic:** Cantidad de datos salientes en una red.

P

- **Post-policy Throughput:** Rendimiento de la red después de la aplicación de políticas de control de tráfico.
- **Pre-policy Throughput:** Rendimiento de la red antes de la aplicación de políticas de control de tráfico.
- **Pre-policy Throughput (packets per second):** Número de paquetes transmitidos por segundo antes de la aplicación de políticas de control de tráfico.
- **Process CPU:** Uso de CPU asociado a un proceso específico en un sistema.
- **Process Count:** Número total de procesos en ejecución en un sistema.
- **Process Memory:** Cantidad de memoria utilizada por un proceso en ejecución.

R

- **Reachability:** Capacidad de acceder a un sistema o servicio dentro de una red.
- **Reads:** Número de operaciones de lectura realizadas en un sistema de almacenamiento.
- **Red Wi-Fi:** Red inalámbrica que utiliza ondas de radio para proporcionar conexión a internet y facilitar la comunicación entre dispositivos sin necesidad de cables físicos.

S

- **Service Level Agreement (SLA):** Acuerdo entre un proveedor de servicios y un cliente que define los niveles esperados de calidad y disponibilidad del servicio, así como las responsabilidades y penalizaciones en caso de incumplimiento.
- **System CPU Usage:** Uso total del procesador en un sistema operativo.
- **System Uptime:** Tiempo total que un sistema ha estado operativo sin reinicios.

T

- **Tecnologías de la Información (TI):** Conjunto de herramientas, procesos y metodologías utilizados para recoger, procesar y distribuir información digital.
- **Técnicas Estadísticas:** Métodos matemáticos utilizados para analizar, interpretar y presentar datos, facilitando la toma de decisiones basada en evidencia cuantitativa.
- **Telecomunicaciones:** Tecnología que permite la transmisión de información a largas distancias a través de medios electrónicos, como teléfonos, redes de computadoras y satélites.

U

- **User CPU Usage:** Cantidad de CPU utilizada por los procesos de usuario en un sistema.

W

- **Writes:** Número de operaciones de escritura realizadas en un sistema de almacenamiento.

Resumen

El objetivo de esta tesis consiste en desarrollar un modelo predictivo para anticipar caídas en access points (APs) a partir del análisis de métricas operativas recolectadas en febrero de 2024 en una red comercial con presencia nacional en México, delimitando así su alcance a este conjunto de datos. Para ello, se construyó una base de datos integrando métricas periódicas de CPU, memoria y tráfico con eventos de caídas de servicio. El preprocesamiento incluyó imputación de valores faltantes, normalización Min-Max y reducción de dimensionalidad mediante UMAP.

Se implementaron tres algoritmos de clasificación: Regresión Logística, Random Forest y Gradient Boosting. La evaluación se realizó con validación cruzada estratificada, midiendo exactitud, sensibilidad y precisión. Los resultados principales muestran una exactitud promedio de 92 % para Regresión Logística, 97 % para Random Forest y 96 % para Gradient Boosting. El análisis de importancia de variables reveló que Process count, Process CPU y System CPU usage son los predictores más relevantes.

Las conclusiones indican que la carga de procesamiento es un factor crítico en la ocurrencia de fallos en APs. Estos hallazgos proveen una base científica para el desarrollo de sistemas de monitoreo proactivo de redes Wi-Fi. Se recomienda validar la generalización de los modelos con datos externos antes de su implementación en otros entornos.

Introducción

En el contexto actual, caracterizado por la creciente digitalización de procesos y servicios, las tecnologías de la información han adquirido un papel central en la vida cotidiana y en el funcionamiento de las organizaciones. Dentro de este panorama, las redes Wi-Fi se han consolidado como un componente esencial de la infraestructura tecnológica, permitiendo la conexión inalámbrica de dispositivos a internet y facilitando el intercambio eficiente de información. Estas redes ofrecen flexibilidad, movilidad y escalabilidad, atributos que han promovido su adopción masiva tanto en entornos corporativos como en espacios públicos y privados.

Los access points (APs) desempeñan un papel fundamental en las redes Wi-Fi, ya que funcionan como nodos intermedios que conectan los dispositivos de los usuarios con la red cableada principal. La calidad del servicio de una red inalámbrica depende en gran medida del rendimiento y la disponibilidad de los APs. Sin embargo, a pesar de los avances tecnológicos y los esfuerzos por parte de fabricantes líderes como Aruba, Huawei y Meraki, los fallos en los APs continúan siendo una realidad persistente. Estos fallos pueden provocar interrupciones en la conectividad, con impactos negativos significativos sobre la productividad de los usuarios, la eficiencia operativa de las organizaciones y la satisfacción general de los clientes. La ausencia de mecanismos proactivos para detectar y anticipar estas fallas limita la capacidad de gestión eficiente de las redes inalámbricas y genera una dependencia reactiva frente a los problemas.

En este sentido, el problema de investigación se puede expresar de manera explícita de la siguiente forma: los fallos en access points (APs) de redes Wi-Fi generan interrupciones en la conectividad, afectando la productividad y la satisfacción del usuario. La falta de estrategias predictivas limita la gestión proactiva de estas fallas.

Frente a esta problemática, la ciencia de datos ofrece herramientas que permiten extraer conocimiento a partir de grandes volúmenes de datos, mediante el uso de técnicas estadísticas, algoritmos de aprendizaje automático y modelos de inteligencia artificial. En particular, los modelos predictivos permiten anticipar eventos futuros con base en datos históricos, lo que los convierte en herramientas valiosas para la detección temprana de anomalías en sistemas complejos como las redes Wi-Fi. La aplicación de modelos de clusterización y clasificación al análisis de métricas operativas de los APs constituye una estrategia prometedora para detectar comportamientos atípicos y predecir fallos antes de que se materialicen.

El principal objetivo, de modo general, de la investigación que aquí se presenta, es el de desarrollar un modelo predictivo con la idea de anticipar fallos en APs utilizando métricas operativas. Para conseguirlo, se plantean, a su vez, objetivos específicos que se muestran a continuación: construir una base de datos con métricas operativas de los access points, implementar algoritmos de clasificación que permitan identificar dispositivos con alta probabilidad de fallo, e identificar las métricas clave asociadas a dichos fallos. La consecución de estos objetivos permitirá avanzar en la construcción de sistemas de mantenimiento predictivo aplicables a la gestión de redes Wi-Fi.

Esta tesis se estructura de la siguiente manera. En el capítulo 1 se presenta el planteamiento general del problema, así como la justificación del estudio y sus alcances. En el capítulo 2 se describe el proceso de construcción y preprocesamiento de la base de datos, incluyendo la recolección, integración y transformación de los datos utilizados. El capítulo 3 aborda el diseño del estudio, explicando los fundamentos teóricos de los algoritmos aplicados, la metodología seguida y el ajuste de los modelos de predicción. Finalmente, en el capítulo 4 se presentan los resultados obtenidos, se discuten sus implicaciones y se proponen líneas futuras de trabajo.

Como punto de partida, esta investigación formula la siguiente hipótesis: las métricas relacionadas con el uso de CPU (por ejemplo, Process Count y Process

CPU) son predictores significativos de fallos en access points. Esta hipótesis no es arbitraria, sino que se basa en la experiencia empírica de algunos operadores de la red con años de experiencia.

Capítulo 1
Generalidades

Capítulo 1. Generalidades

Antes de abordar el planteamiento específico del problema, es necesario contextualizar el entorno en el que se enmarca esta investigación. La convergencia entre las redes inalámbricas y la ciencia de datos ha abierto nuevas posibilidades para la gestión inteligente de infraestructuras tecnológicas. En este sentido, el presente trabajo se posiciona en la intersección entre ambos campos, con el objetivo de explorar el potencial de los modelos predictivos para anticipar fallos en dispositivos críticos como los access points, los cuales desempeñan un papel esencial en la conectividad de redes Wi-Fi en entornos comerciales.

1.1 Planteamiento del problema

Los Access Points (APs) constituyen un componente fundamental para garantizar la conectividad inalámbrica en redes Wi-Fi. Estos dispositivos permiten la conexión entre los usuarios finales y la infraestructura de red, siendo esenciales en espacios amplios como tiendas departamentales, oficinas corporativas y entornos educativos. Sin embargo, la presencia de fallos en APs puede generar interrupciones críticas en el servicio, afectando la productividad, la experiencia del usuario y, en contextos comerciales, la satisfacción del cliente. A pesar de los avances tecnológicos en el diseño y manufactura de estos dispositivos por parte de empresas como Aruba, Huawei y Meraki, la posibilidad de fallos operativos persiste, lo que subraya la necesidad de mecanismos más eficientes para su gestión preventiva.

La relevancia del problema radica en que las fallas en APs no solo representan un obstáculo técnico, sino que conllevan consecuencias económicas y operativas significativas. El mantenimiento correctivo, aplicado posterior al fallo, suele implicar pérdidas por tiempo de inactividad y un uso ineficiente de los recursos técnicos. Frente a esta situación, surge la necesidad de transitar hacia un enfoque

proactivo, donde la detección temprana de anomalías permita intervenir antes de que ocurra una interrupción del servicio.

En este contexto, la ciencia de datos ofrece herramientas útiles para abordar el problema desde una perspectiva predictiva. La disponibilidad de métricas operativas registradas de forma continua por los APs abre la posibilidad de aplicar modelos de aprendizaje automático que permitan anticipar comportamientos anómalos o indicios de falla. Sin embargo, en la literatura científica, aunque existen trabajos que aplican técnicas de ciencia de datos para el análisis del comportamiento de redes o el estudio del flujo de usuarios conectados a Wi-Fi, son escasos los estudios centrados específicamente en la predicción de fallos en APs a partir de sus métricas operativas.

Esta brecha en la literatura representa una oportunidad para explorar e implementar modelos de machine learning que aprovechen datos históricos de rendimiento de APs con el objetivo de prever fallas inminentes. Los estudios más cercanos al problema, como Network Equipment Failure Prediction with Big Data Analytics (2020), Failure Prediction Using Machine Learning and Time Series in Optical Networks (2019), y Predicting LAN Switch Failures: An Integrated Approach with DES–SVM (2021), abordan problemas similares, pero en diferentes tipos de dispositivos y con enfoques variados. Estas investigaciones evidencian el potencial de las técnicas predictivas, aunque sin centrarse en APs de redes Wi-Fi, lo que refuerza la originalidad y pertinencia del presente estudio.

Por tanto, el problema central que se plantea en esta investigación es el siguiente: ¿es posible anticipar fallos en access points mediante el análisis de sus métricas operativas utilizando técnicas de aprendizaje automático, y con ello optimizar la gestión de redes Wi-Fi en entornos de alta demanda? La resolución de este problema permitiría transitar hacia modelos de mantenimiento predictivo que mejoren la eficiencia operativa y reduzcan las interrupciones del servicio, aportando valor tanto técnico como económico a las organizaciones que dependen de una conectividad inalámbrica estable.

A continuación, se enlista literatura revisada, se encontraron algunos trabajos, los tres más relevantes son:

- Lam, H. S., Tan, Y. F., Soo, W. K., Guo, X., & Lee, Z. M. (2016). Network equipment failure prediction with big data analytics. International Journal on Soft Computing Applications, 8(3), 1–15.
- Wang, Z., Zhang, M., Wang, D., Song, C., Liu, M., Li, J., Lou, L., & Liu, Z. (2019). Failure prediction using machine learning and time series in optical network. Optics Express, 27(3), 10–12.
- Myrzatay, A., Rzayeva, L., Bandini, S., Shaye, I., Saoud, B., Çolak, I., & Kayisli, K. (2021). Predicting LAN switch failures: An integrated approach with DES and machine learning techniques (RF/LR/DT/SVM). IEEE Transactions on Network and Service Management, 18(2), 100–110.

1.2 Protocolo de investigación

Esta investigación se desarrolla bajo un enfoque cuantitativo, utilizando técnicas de aprendizaje supervisado para la predicción de fallos en Access Points (APs). El diseño del estudio contempla la construcción de una base de datos a partir de registros históricos de métricas operativas extraídas periódicamente de los APs, así como eventos etiquetados como fallos. Posteriormente, se entrenarán y evaluarán modelos de clasificación para predecir el tipo de fallo que puede presentarse en la siguiente hora.

La metodología general incluye las siguientes etapas:

1. Adquisición y construcción de la base de datos: Se describe a profundidad en 2.1 y 2.2

2. Preprocesamiento de datos: Se lleva a cabo limpieza de valores atípicos, manejo de valores faltantes, normalización y reducción de dimensionalidad.
3. Definición de la variable objetivo: La variable dependiente es el tipo de fallo que ocurre en un Access Point durante la siguiente hora, categorizada con base en los eventos registrados. Se opta por un enfoque binario (0: sin fallo, 1: con fallo) en la etapa de modelado inicial.
4. Selección de métricas (variables independientes): Las métricas consideradas provienen de registros técnicos generados por los APs, y fueron seleccionadas con base en su disponibilidad, frecuencia de actualización y relación potencial con el estado operativo del dispositivo.
5. Entrenamiento y evaluación de modelos: Se emplean algoritmos, particularmente de clasificación supervisada, por ejemplo: máquinas de soporte vectorial (SVM por sus siglas en inglés), regresión logística, árboles aleatorios y finalmente «gradient boosting». Los modelos se evalúan mediante validación cruzada y métricas como precisión, recall, F1-score y matriz de confusión.
6. Análisis de importancia de variables: Para identificar qué métricas se asocian más fuertemente con la ocurrencia de fallos, se estudia el impacto de las variables en estos modelos y se entrena un árbol de decisión de baja profundidad como herramienta interpretativa.

A continuación, se presenta la tabla que resume las variables consideradas en el estudio:

Variable	Tipo	Descripción
Process CPU	Independiente	Porcentaje de CPU utilizado por procesos específicos
Process count	Independiente	Número total de procesos activos en el dispositivo
Process memory	Independiente	Memoria consumida por los procesos del sistema
System CPU usage	Independiente	Porcentaje total de CPU utilizado por el sistema
User CPU usage	Independiente	Porcentaje de CPU utilizado por procesos del usuario

CPU usage	Independiente	Uso combinado del CPU en todas las funciones del dispositivo
Load average	Independiente	Promedio de carga del sistema durante intervalos recientes
Memory usage	Independiente	Porcentaje de memoria total utilizada
Dropped throughput	Independiente	Volumen de datos descartados por congestión o errores
Latency	Independiente	Tiempo de respuesta promedio del dispositivo
Availability	Independiente	Porcentaje de tiempo en que el dispositivo estuvo operativo
Fallo de AP	Dependiente	Evento binario (0: sin fallo, 1: con fallo)

Tabla 1. Variables del estudio

1.3 Justificación

La presente investigación se justifica por la necesidad de reducir interrupciones en redes Wi-Fi empresariales, cuyo funcionamiento estable depende en gran medida del desempeño de los Access Points (APs). Estos dispositivos actúan como nodos críticos en la infraestructura de conectividad, y su mal funcionamiento puede derivar en pérdidas económicas, deterioro en la experiencia del usuario y afectaciones operativas para las organizaciones que dependen de la conectividad continua, como es el caso de tiendas departamentales, centros logísticos o establecimientos de atención al cliente.

En contextos donde existen acuerdos de nivel de servicio (Service Level Agreement, SLA), los proveedores de Internet están obligados a mantener ciertos estándares de disponibilidad y desempeño. El incumplimiento de dichos acuerdos suele traducirse en sanciones económicas y pérdida de confianza por parte de los clientes. Por lo tanto, resulta estratégico contar con herramientas que permitan anticipar fallos antes de que ocurran, habilitando esquemas de mantenimiento proactivo o acciones correctivas oportunas.

Este estudio propone la creación de modelos de aprendizaje automático para predecir fallos en APs a partir de métricas operativas extraídas en tiempo real. La implementación de dichos modelos no solo mejora la gestión técnica de la red, sino que representa una contribución relevante para administradores de red, empresas usuarias de servicios de conectividad y proveedores de servicios de Internet, al ofrecer un sustento para la oportuna toma de decisiones fundamentadas en datos. Además, se aporta al conocimiento sobre el uso de técnicas para la predicción de fallos en sistemas de telecomunicaciones, área que aún presenta oportunidades de exploración y mejora.

1.4 Límites y alcances

Este estudio presenta límites claramente definidos en cuanto al tiempo, espacio y tipo de evento analizado. Temporalmente, se restringe al análisis de los datos recolectados durante el mes de febrero de 2024. Esta delimitación responde a la necesidad de acotar el volumen de datos para facilitar su análisis y garantizar un enfoque detallado. No obstante, se reconoce que las condiciones técnicas, de tráfico o de configuración de red pueden variar en otros periodos del año, por lo que los resultados podrían no generalizarse automáticamente a meses distintos. Para abordar esta limitación, en futuras fases del proyecto se contempla realizar pruebas con datos de otros meses, lo cual permitiría evaluar la estabilidad del modelo predictivo ante variaciones temporales y mejorar su capacidad de generalización.

En términos espaciales, el estudio se enfoca exclusivamente en una tienda con presencia nacional en México, pero con datos de una única ubicación física. Esta decisión facilita el control de variables contextuales y permite un análisis más homogéneo, aunque también reduce la representatividad de los resultados frente a otras tiendas o regiones del país. Para mitigar esta limitación, se propone que una siguiente etapa del estudio considere datos de distintas sucursales, lo que permitiría ajustar el modelo a condiciones técnicas diversas y robustecer su aplicabilidad.

Respecto al alcance temático, el proyecto se concentra únicamente en la predicción de caídas del servicio en los access points, excluyendo otras formas de degradación del desempeño como la pérdida de paquetes, la latencia elevada o el jitter. Esta delimitación permite enfocar los recursos computacionales y analíticos en un tipo de evento crítico para la operación empresarial, pero implica que el modelo desarrollado no detectará otras fallas potenciales. Para ampliar su utilidad, se planea en trabajos futuros extender la clasificación a otros tipos de eventos de red, lo que permitiría una gestión más integral de la infraestructura inalámbrica.

Pese a estas limitaciones, el estudio tiene un alcance significativo dentro de su contexto de aplicación. El modelo desarrollado se orienta a la detección temprana de caídas del servicio en access points, lo cual tiene un efecto evidentemente directo en la parte operativa de la compañía, al permitir acciones preventivas que reduzcan el tiempo de inactividad, las penalizaciones contractuales y las afectaciones al cliente. Asimismo, el conocimiento derivado del análisis de métricas específicas contribuirá en que las decisiones de los administradores de red sean informadas por. Finalmente, aunque nuestro modelo se ha entrenado mediante información recabada de febrero de 2024 y de una tienda en particular, su diseño permite ser escalado y adaptado a otras temporalidades y ubicaciones mediante ajustes en las variables de entrada, lo que amplía su potencial de uso a nivel corporativo.

Capítulo 2
Base de datos

Capítulo 2. Base de datos

En este capítulo, describimos las características y la obtención del archivo base de este trabajo, que fue generado a través de dos tablas, una con eventos y otra con métricas de dispositivos, esta labor tuvo las siguientes partes:

Adquisición y preprocesamiento de datos: Se utilizó un conjunto de datos almacenado en BigQuery, el cual fue consultado mediante SQL y posteriormente procesado con Pandas. La base de datos integra métricas extraídas de dispositivos de red y eventos registrados, filtrando únicamente aquellos relacionados con umbrales de ancho de banda excedidos. Se realizaron transformaciones para eliminar valores irrelevantes, estandarizar las métricas y definir la variable objetivo.

Reducción de dimensionalidad: Se implementaron cinco métodos de reducción de dimensionalidad: t-Distributed Stochastic Neighbor Embedding (t-SNE), Singular Value Decomposition (SVD) que es el que se muestra en la **Figura 1**, Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF) y Sparse Matrix Factorization (SMF). Estos algoritmos permitieron representar los datos en un espacio de menor dimensión, facilitando la visualización y la identificación de patrones de agrupamiento en los datos.

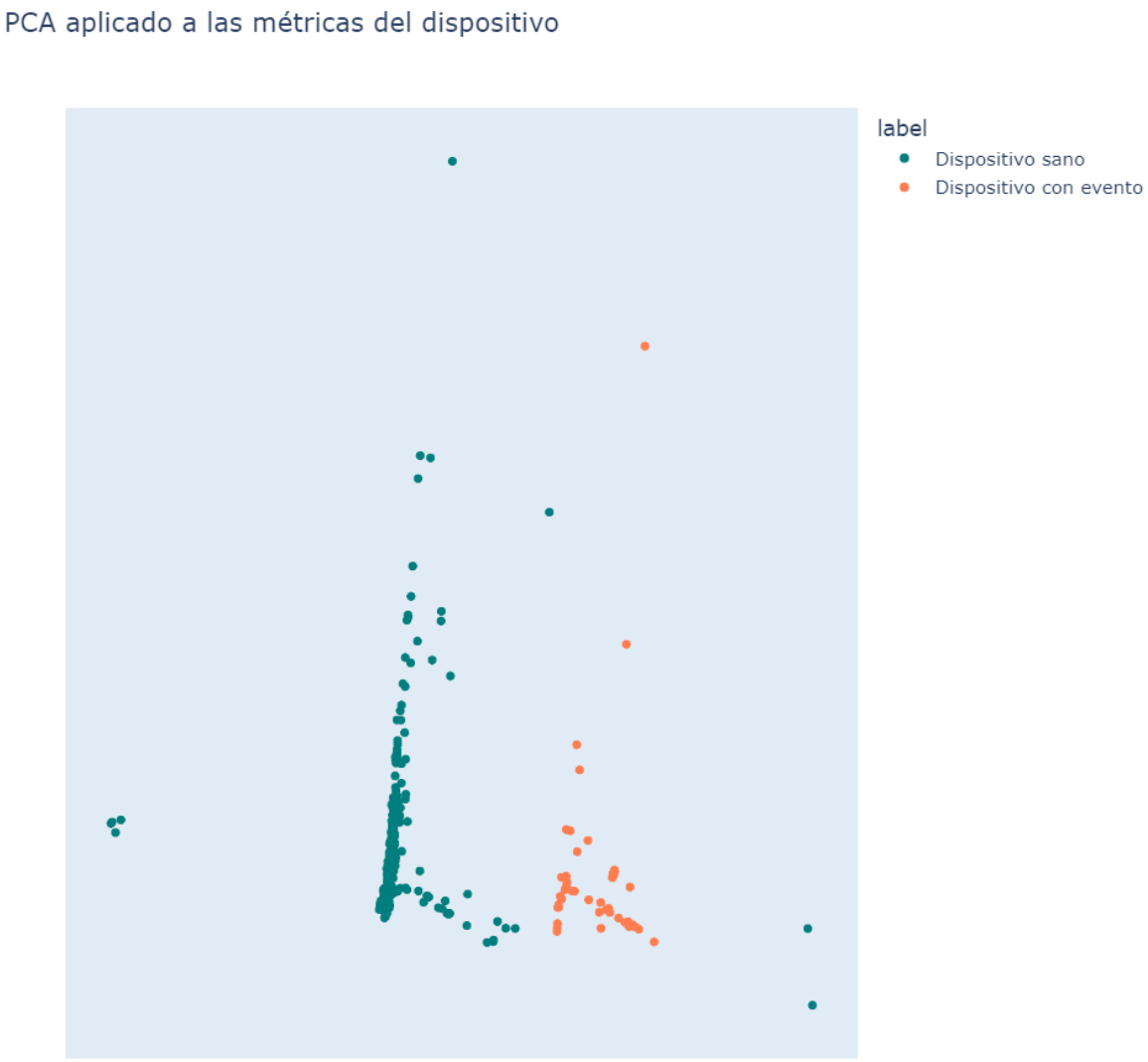

Figura 1: Reducción de dimensión con PCA. Fuente: Elaboración propia

Definición de la variable objetivo y división del conjunto de datos: La variable de clasificación se definió de forma binaria, donde la categoría "No event" se codificó como 0 y cualquier otro evento como 1. Posteriormente, se realizó la partición del conjunto de datos en entrenamiento y prueba, con una proporción del

70 % y 30 %, respectivamente, asegurando el mantenimiento de la distribución de clases mediante estratificación.

2.1 Construcción de la base de datos

Como se menciona al inicio os datos fueron obtenidos desde Big Query, de Google, a través de una empresa de monitoreo de infraestructura de red (Sopris Technologies), y cubren el período de febrero de 2024.

No se ha utilizado un sistema gestor de bases de datos, puesto que directamente se ha empleando Python 3, junto con la biblioteca Pandas, para la manipulación y análisis de los datos. Esto se debe a la facilidad y seguridad que implica el trabajar con una copia local de los datos y no en el entorno productivo de Sopris Technologies, esto es posible debido a que la cantidad de datos no es masiva y permite aislar la investigación sin la posibilidad de generar gastos en consultas en Big Query y sin depender de las credenciales de acceso.

Para ilustrar el contenido de los datos, se muestran las columnas presentes en este DataFrame. La segunda columna corresponde al evento, representado por un número entre 0 y 33, ya que existen 33 eventos diferentes, incluyendo el caso «no evento», que es cuando el dispositivo no presenta un evento. Cada número codifica un evento específico. Además, las columnas incluyen las métricas recopiladas por los dispositivos antes de que ocurriera un evento determinado o, en su defecto, si no se presentó ningún evento, como se muestra en la Tabla 2:

Tipo	Nombre de la métrica/evento
ID	device_id
Evento	Access points down threshold exceeded
Métrica	Aggregate Fragment Throughput
Métrica	Availability
Métrica	CPU usage

Métrica	Dropped throughput
Métrica	Dropped throughput (packets per second)
Métrica	IO Memory usage
Métrica	Inbound discards
Métrica	Inbound errors
Métrica	Inbound traffic
Métrica	Latency
Métrica	Load average
Métrica	Memory usage
Métrica	No buffer dropped throughput (packets per second)
Métrica	Operational status
Métrica	Outbound discards
Métrica	Outbound errors
Métrica	Outbound traffic
Métrica	Post-policy throughput
Métrica	Pre-policy throughput
Métrica	Pre-policy throughput (packets per second)
Métrica	Process CPU
Métrica	Process count
Métrica	Process memory
Métrica	Reachability
Métrica	Reads
Métrica	System CPU usage
Métrica	System uptime
Métrica	User CPU usage
Métrica	Write

Tabla 2: Columnas del dataframe

Para generar el conjunto de datos a partir de las métricas y los eventos, se sigue un proceso sistemático utilizando la biblioteca pandas en Python (Ver creator-dataframe.py), como se ve en la Figura 2.

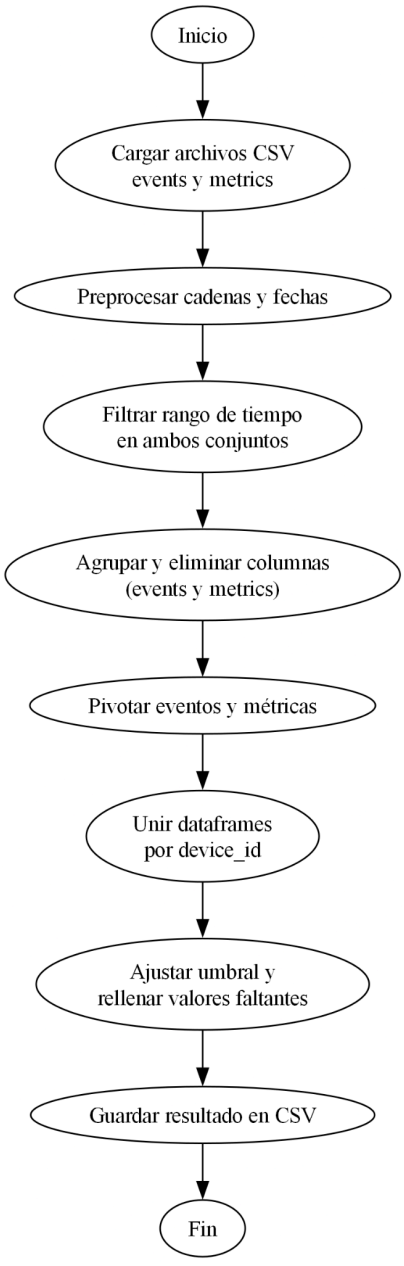

Figura 2: Procesamiento de datos. Fuente: Elaboración propia

Para la creación de la base de datos utilizada en este trabajo, se partió de dos fuentes principales: un archivo con eventos reportados por los Access Points (los 32 mencionados anteriormente) y otro con las métricas de desempeño recolectadas por los mismos dispositivos (mostradas explícitamente arriba). Ambos archivos, en formato CSV, fueron cargados en estructuras de datos tipo DataFrame mediante la librería pandas.

A continuación, se realizó un preprocesamiento de ambas fuentes. En el caso del archivo de métricas, se modificaron las etiquetas de las métricas para diferenciarlas claramente y se convirtió la columna de marcas de tiempo al formato datetime. De manera similar, en el archivo de eventos, se ajustaron las etiquetas de clasificación para distinguirlas como eventos y también se transformaron las fechas al mismo formato, redondeándolas al segundo más cercano para mantener consistencia temporal.

Una vez estandarizados los formatos, se filtraron los datos de ambas fuentes para acotar el análisis a un intervalo temporal específico, definido por una fecha de referencia y una duración en minutos**.** Este paso permitió acotar el volumen de datos a procesar y enfocarse en los eventos que ocurrieron en un periodo determinado, junto con las métricas correspondientes recogidas al inicio de ese intervalo.

Luego de este filtrado temporal, se eliminaron las columnas de tiempo y se agruparon los datos por identificador de dispositivo y tipo de métrica o evento. En el caso de los eventos, se sumó el número de incidencias por clasificación, mientras que para las métricas se calculó el promedio registrado por cada AP durante el periodo observado.

Posteriormente, se reorganizaron los datos agrupados mediante operaciones de pivotado. Esto permitió convertir las clasificaciones de eventos y tipos de métricas en columnas individuales, lo cual facilitó la integración de los datos. El resultado fue un DataFrame en el que cada fila representaba a un AP y cada columna correspondía a una métrica o evento. Finalmente, se realizó una fusión entre

ambos DataFrames —el de eventos y el de métricas— empleando el identificador del dispositivo como llave común. Se ajustó la variable objetivo, que fue explicado en el resumen inicial.

2.1.1 Métricas clave para evaluar Access Points

A continuación, se detallan las métricas clave que se deben monitorear para evaluar el desempeño de los AP:

Intensidad de la señal (RSSI): El Indicador de Intensidad de Señal Recibida (RSSI, por sus siglas en inglés) mide la potencia de la señal que un dispositivo recibe del AP. Valores más altos indican una mejor calidad de conexión, mientras que valores bajos pueden señalar problemas de cobertura o interferencia.

Ancho de banda disponible: Esta métrica indica la capacidad máxima de transmisión de datos que el AP puede manejar en un momento dado. Un ancho de banda adecuado es crucial para soportar múltiples dispositivos y aplicaciones que demandan altas tasas de transferencia de datos.

Tasa de transferencia de datos: Refleja la velocidad a la que se transmiten los datos entre el AP y los dispositivos conectados. Velocidades de transferencia consistentes y altas son indicativas de un buen rendimiento del AP.

Tasa de errores de paquetes: Esta métrica mide el porcentaje de paquetes de datos que no se transmiten correctamente y deben ser retransmitidos. Una tasa alta de errores puede indicar interferencias, problemas de hardware o congestión en la red.

Latencia: La latencia se refiere al tiempo que tarda un paquete de datos en viajar desde el dispositivo emisor hasta el receptor. Bajas latencias son esenciales para aplicaciones en tiempo real, como videoconferencias o juegos en línea.

Jitter: El jitter mide la variabilidad en el tiempo de llegada de los paquetes de datos. Altos niveles de jitter pueden causar interrupciones en aplicaciones sensibles al tiempo, como VoIP o streaming de video.

Número de dispositivos conectados: Monitorear la cantidad de dispositivos conectados a un AP ayuda a identificar posibles sobrecargas y a planificar la capacidad de la red de manera efectiva.

Uso de CPU y memoria del AP: Evaluar el consumo de recursos del AP permite detectar posibles cuellos de botella o necesidades de actualización de hardware.

Al monitorear estas métricas, los administradores de red pueden identificar áreas de mejora, optimizar el rendimiento de los Access Points y asegurar una experiencia de usuario satisfactoria en la red inalámbrica. Estas métricas son las que usaremos para tratar de predecir las caídas en los APs

2.2 Preprocesamiento de la base de datos

Durante la construcción de la base de datos, se aplicaron técnicas de preprocesamiento con el objetivo de garantizar la calidad y la homogeneidad de los datos antes de entrenar los modelos de clasificación. Este proceso incluyó la imputación de valores perdidos y la normalización de atributos, ambas etapas fundamentales al tratarse de registros recolectados a lo largo del tiempo por dispositivos con diferentes características y condiciones de operación.

En primer lugar, la imputación de valores perdidos se abordó considerando la naturaleza temporal de los datos. Se optó por imputar el valor faltante utilizando el promedio de las dos observaciones más cercanas en el tiempo, una anterior y una posterior, siempre que estuvieran disponibles. Esta decisión se basó en la suposición de continuidad temporal en las métricas operativas de los Access Points. A diferencia de otras estrategias como la imputación por mediana o por valor constante, este método conserva la tendencia local del dato dentro de su contexto temporal, sin introducir discontinuidades artificiales.

En segundo lugar, se aplicó la normalización Min-Max a todos los atributos numéricos. Esta técnica transforma los valores originales de cada variable a un rango comprendido entre 0 y 1, manteniendo las relaciones proporcionales entre observaciones. La elección de este método responde a la necesidad de evitar que variables con escalas mayores —por ejemplo, el uso de memoria en bytes— dominen el proceso de clasificación frente a otras variables con menor rango —como la intensidad de señal en decibelios—. Otras técnicas como la estandarización Z-score podrían haber sido consideradas, pero Min-Max resultó más apropiada en este caso debido a su capacidad de preservar la distribución original en problemas donde los datos están acotados de forma natural y se busca mantener la interpretabilidad de las magnitudes relativas.

El proceso completo de preprocesamiento puede resumirse en el siguiente pseudocódigo:

```
Algoritmo: Preprocesamiento de datos
1. Cargar archivos CSV de eventos y métricas.
2. Para cada métrica en el conjunto de datos:
   a. Si el valor está ausente:
      i. Reemplazar con el promedio del anterior y posterior.
3. Para cada atributo numérico:
   a. Calcular el valor mínimo y máximo de la variable.
   b. Aplicar normalización Min-Max:
      valor_normalizado = (valor - mínimo) / (máximo - mínimo)
4. Exportar el DataFrame limpio a un nuevo archivo CSV.
```

Estas etapas fueron implementadas antes de aplicar los algoritmos de reducción de dimensionalidad y clasificación, garantizando que los modelos partieran de datos comparables y sin sesgos derivados de diferencias de escala o ausencias.

2.3 Procesamiento y Filtrado de Datos

2.3.1 Necesidad Abordada

Como se vio en 2.1, la base de datos contenía dos tipos principales de tablas: eventos y métricas de los Access Points. La intervención fue necesaria debido a la gran cantidad de datos estructurados y la necesidad de filtrar eventos específicos dentro de un rango de tiempo para el análisis. Además, era crucial combinar estas dos bases de datos para facilitar el análisis conjunto de eventos y métricas. Como se verá en 2.3.3, se filtraron los eventos etiquetados como “Access points down threshold exceeded", puesto que son que corresponden a caídas de dispositivos, que es lo que busca predecir este trabajo.

2.3.2 Algoritmo o Estrategia Utilizada

Se utilizaron funciones en para cargar, filtrar y procesar los datos. Se implementó la función `procesar_df`, que realiza las tareas mostradas en la Figura 3:

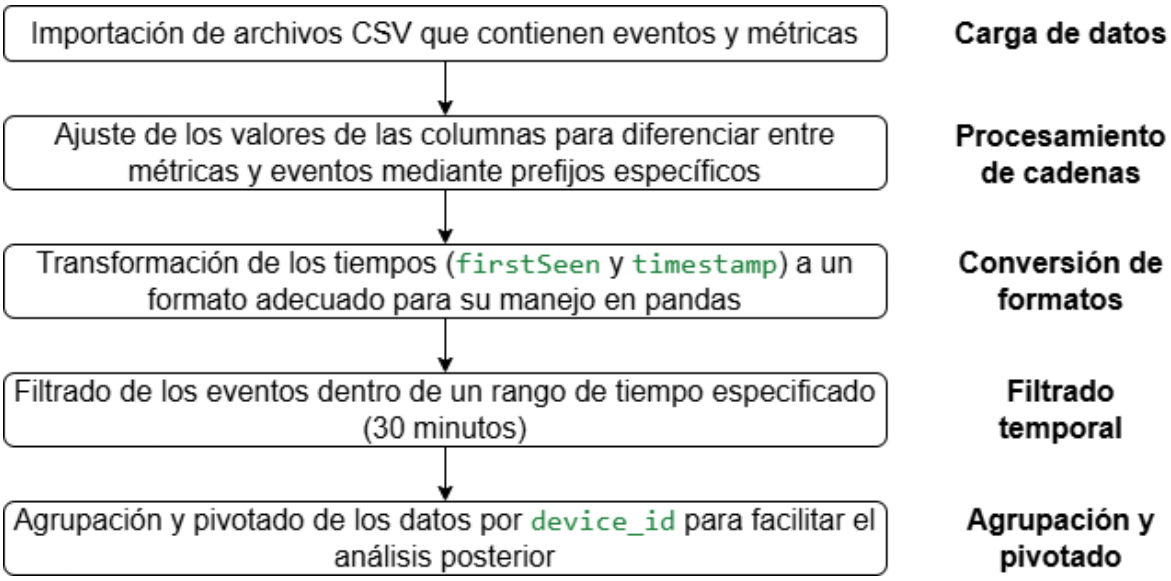

Figura 3: Flujo de datos. Fuente: Elaboración propia

El código correspondiente se puede ver en el Anexo 1.

2.3.3 Análisis Estadístico y Gráfico Antes del Procesamiento

- **Datos de Eventos**: Los datos estaban dispersos con múltiples tipos de eventos no relacionados y una alta densidad de eventos irrelevantes (como errores de monitoreo).
- **Datos de Métricas**: Los datos de métricas estaban sin agrupar y contenían múltiples entradas por dispositivo, complicando la interpretación.
- **Pivotado y Agrupación**: La base de datos resultante fue más estructurada, con cada fila representando un dispositivo único y sus métricas relevantes.
- **Gráfico de Correlación**

2.4 Análisis exploratorio de los datos

Una de las métricas más relevantes para medir la salud de un dispositivo de red es el uso de CPU, a continuación se muestra un histograma del uso de CPU (Figura 4):

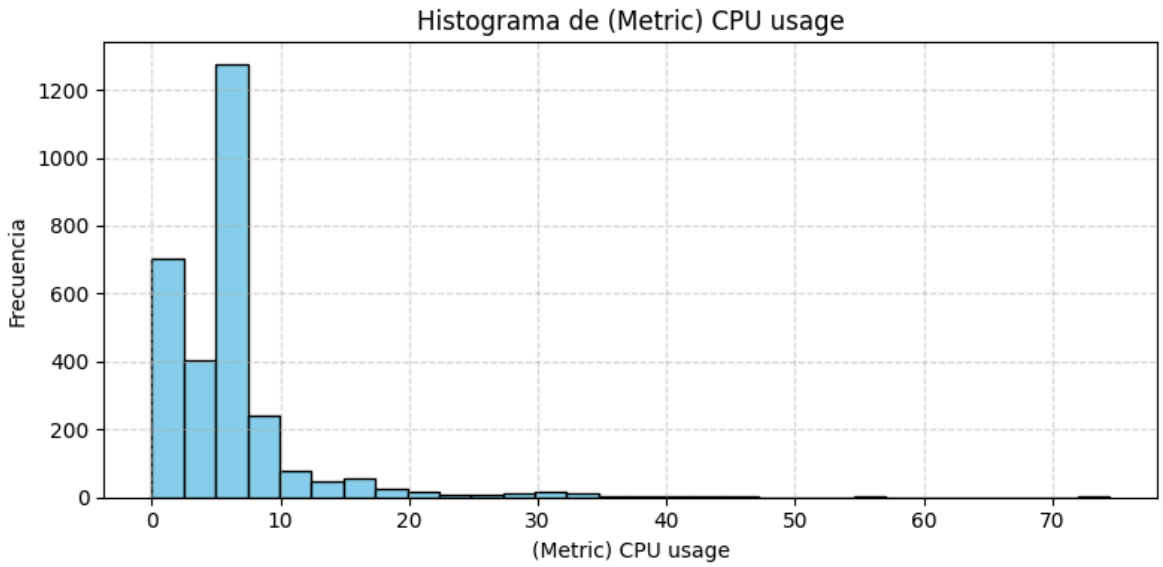

Figura 4: Histograma de uso de CPU. Fuente: Elaboración propia

A continuación, se muestra la matriz de correlación que fue el inicio de este trabajo, donde se mezclan métricas con eventos y sus coeficientes de correlación

de Pearson, para detectar dependencias lineales entre ellas, como se ve en la Figura 5, donde se observa fuerte correlación entre métricas.

Figura 5: Matriz de correlación. Fuente: Elaboración propia

Capítulo 3

Diseño del estudio y ajuste de modelos

Capítulo 3. Diseño del estudio y ajuste de modelos

El capítulo 2 presentó la construcción de la base de datos, desde la recopilación y transformación de las métricas de los dispositivos hasta el análisis exploratorio que permitió identificar patrones iniciales y características relevantes. A partir de esta estructura de datos consolidada, es posible avanzar hacia el diseño y ajuste de los modelos de machine learning, cuyo propósito central es predecir fallos en los Access Points a partir de sus métricas operativas.

El capítulo 3 desarrolla este proceso, estableciendo primero un marco teórico que describe los algoritmos y técnicas clave a emplear, seguido de la metodología que define las etapas concretas desde la preparación de los datos hasta la validación de los modelos. Finalmente, se detallan los procedimientos y criterios aplicados para ajustar los modelos seleccionados —Regresión Logística, Random Forest y Gradient Boosting—, optimizando sus parámetros y garantizando su rendimiento en la detección anticipada de fallos.

Esta transición del análisis exploratorio a la modelización marca el paso de entender la estructura de los datos a aprovecharla para generar predicciones útiles y precisas, alineadas con el objetivo de mejorar la gestión proactiva de la red.

3.1 Marco teórico y metodológico

A continuación se presenta la sección de antecedentes y una tabla de discusión comparativa, la cual integra estudios previos relevantes y expone de forma comparativa los enfoques y hallazgos de dichos trabajos con el presente estudio.

Antecedentes

Diversos trabajos han abordado la predicción de fallos en equipos de red y la correlación entre eventos de inalcanzabilidad y las métricas de desempeño, aunque con enfoques y metodologías diferenciadas *(Gill, Jain, & Nagappan, 2011; Network Equipment Failure Prediction with Big Data Analytics, 2020; Failure Prediction Using Machine Learning and Time Series in Optical Networks, 2019)*. Por ejemplo, se han realizado análisis a gran escala en redes de centros de datos para evaluar la fiabilidad de dispositivos y la efectividad de la redundancia en la reducción del impacto de fallos *(Gill, Jain, & Nagappan, 2011)*. En otros estudios se ha utilizado el análisis de grandes volúmenes de datos mediante técnicas de regresión y algoritmos de aprendizaje automático, alcanzando altos niveles de precisión en la predicción de fallos hasta días antes de su ocurrencia *(Network Equipment Failure Prediction with Big Data Analytics, 2020)*. De igual forma, investigaciones en redes ópticas han empleado métodos que combinan máquinas de soporte vectorial y análisis de series temporales para anticipar eventos de mal rendimiento *(Failure Prediction Using Machine Learning and Time Series in Optical Networks, 2019; Predicting LAN Switch Failures: An Integrated Approach with DES–SVM, 2021)*. Estos antecedentes proveen un marco de referencia que resalta la importancia de identificar y correlacionar métricas operativas con la ocurrencia de fallos, aspecto central en el presente trabajo.

Discusión Comparativa

El objetivo de este trabajo se orienta a correlacionar los eventos de inalcanzabilidad de los dispositivos de red con sus métricas de desempeño, apoyándose en la integración de técnicas de reducción de dimensionalidad y algoritmos de clasificación. En la siguiente tabla se presenta una comparación entre tres estudios relevantes y el enfoque adoptado en este trabajo:

Referencia	Problema	Metodología	Resultados	Similitudes con este estudio
Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications	Correlacionar eventos de dispositivos no alcanzables con sus métricas de desempeño.	Análisis de eventos SNMP/syslog, tráfico y topología para identificar fallas reales y su impacto en el tráfico.	Análisis a gran escala en centros de datos; identifica que la redundancia de red reduce el impacto de fallas hasta en un 40%.	Coincide en la correlación entre eventos de inalcanzabilidad y métricas de rendimiento, aportando insights para mejorar la fiabilidad de la red.
Network Equipment Failure Prediction with Big Data Analytics	Predicción de fallos en equipos de red mediante análisis de big data.	Extracción de patrones desde syslogs y quejas, entrenamiento de un modelo de regresión lineal sobre Hadoop.	Usa reglas y regresión sobre Hadoop, logrando hasta 99.9% de precisión en la predicción de fallos dentro de los 4 días previos.	Alineado con el análisis de grandes volúmenes de datos para identificar métricas clave predictivas y generar alertas tempranas.
Failure Prediction Using Machine Learning and Time Series in Optical Network	Predicción de fallos en equipos ópticos usando SVM y series temporales.	Uso de SVM con kernel RBF y DES para anticipar valores de métricas y predecir fallos en redes ópticas.	Combina SVM y análisis de series temporales (método DES-SVM), logrando una precisión promedio del 95% para anticipar fallos.	Similar enfoque proactivo usando análisis inteligente de métricas operativas y aprendizaje automático para anticipar mal rendimiento.
Fault Detection in Telecom Networks using Bi-level Federated Graph Neural Networks	Detección de fallas en nodos RAN de redes 4G/5G considerando restricciones de privacidad.	Modelo federado con grafos a dos niveles: uno entre nodos RAN y otro en el grafo de ejecución de software de cada nodo.	El modelo personalizado superó a métodos centralizados y federados estándar en detección de fallas reales en nodos RAN.	Utiliza métricas operativas multivariadas para anticipar fallos, similar a la correlación entre métricas y eventos en Access Points.
Fault Detection in Mobile Networks Using Diffusion Models	Detección de fallas silenciosas en redes móviles a partir de métricas de desempeño de software.	Uso de modelos de difusión (SSSDS4) en series de tiempo multivariadas; detección basada en reconstrucción y pronóstico.	El modelo de reconstrucción logró F1 = 0.591, superando a LSTM y GNN en datos reales de nodos RAN.	Utiliza series temporales multivariadas para anticipar fallos, como en la predicción de eventos en Access Points a partir de métricas.
Leveraging Machine Learning for Anomaly Detection in Telecom Network Management	Identificación temprana de fallos en redes telecomunicacionales con alto volumen y complejidad.	Revisión de enfoques supervisados, no supervisados y series de tiempo; propone una arquitectura modular con ML integrado para mantenimiento y monitoreo.	Reporta mejoras de hasta 96% en precisión y reducción del MTTR en 71% con detección temprana basada en ML.	Emplea métricas multivariadas, detección proactiva y aprendizaje supervisado/no supervisado como base para prevenir fallos de red.

Tabla 3: Comparación de estudios relevantes

Como se observa en la **Tabla 3**, mientras que el primer estudio se centra en la evaluación de la fiabilidad mediante el análisis de redundancia de red, los dos siguientes se orientan hacia la predicción anticipada de fallos utilizando técnicas de análisis de grandes volúmenes de datos y métodos de aprendizaje automático. El presente trabajo se diferencia al integrar diversas técnicas de reducción de

dimensionalidad (incluyendo UMAP) y modelos de clasificación, con el objetivo de correlacionar de manera precisa los eventos de inalcanzabilidad con métricas de desempeño, lo que permitirá implementar acciones preventivas y optimizar la gestión de redes Wi-Fi. Esta comparación evidencia la contribución del enfoque adoptado, que combina análisis de datos en profundidad con una metodología de validación robusta, para lograr resultados aplicables en entornos reales de operación.

Algoritmos empleados

Para garantizar un desempeño óptimo en la predicción de fallos en Access Points, se llevó a cabo un proceso de ajuste y evaluación de modelos basado en técnicas de validación cruzada y balanceo de clases. Se consideraron tres algoritmos de clasificación: Regresión Logística, Random Forest y Gradient Boosting, cada uno con configuraciones específicas para maximizar su rendimiento en un conjunto de datos caracterizado por un fuerte desequilibrio entre clases.

La Regresión Logística fue configurada con un número máximo de iteraciones de 1000 y la opción de pesos balanceados para compensar la baja representación de la clase minoritaria. Se utilizó validación cruzada estratificada de cinco particiones, con la métrica de exactitud como criterio de evaluación. En cada fold, el modelo fue entrenado y evaluado en un subconjunto diferente del conjunto de entrenamiento, asegurando que la proporción de clases se mantuviera estable a lo largo del proceso.

El modelo de Random Forest fue ajustado con 100 árboles y la opción de pesos balanceados para manejar el desbalance de clases. La validación cruzada permitió evaluar su capacidad predictiva y ajustar su hiperparámetro principal, el número de estimadores, garantizando una adecuada generalización en la clasificación de eventos. La estabilidad observada en los resultados sugiere que el modelo logra una segmentación precisa de las observaciones con una varianza mínima entre los diferentes folds.

Debido a que Gradient Boosting no admite directamente la ponderación de clases, se empleó un cálculo de pesos de muestra para cada fold durante la validación cruzada. En cada iteración, se ajustaron los pesos en función de la distribución de clases dentro del subconjunto de entrenamiento, permitiendo al modelo aprender patrones sin verse afectado por el desbalance en los datos. El número de estimadores se fijó en 100, asegurando un equilibrio entre rendimiento y eficiencia computacional.

Los modelos fueron comparados en términos de su exactitud promedio obtenida durante la validación cruzada. Mientras que la Regresión Logística presentó una variabilidad significativa en los diferentes folds, los modelos basados en árboles mostraron un desempeño más estable, con Random Forest y Gradient Boosting alcanzando valores de exactitud cercanos al 100 %. La elección del mejor modelo se realizó con base en su capacidad para identificar correctamente los eventos de fallos sin comprometer la generalización a nuevos datos.

3.1.2 Algoritmos de reducción

La reducción de dimensionalidad es una técnica esencial en el análisis de datos que busca simplificar conjuntos de datos de alta dimensión, manteniendo la mayor cantidad de información relevante posible. Esto facilita la visualización, el procesamiento y la interpretación de los datos, además de mejorar la eficiencia de los algoritmos de aprendizaje automático. A continuación, se describen algunos de los algoritmos más destacados en este ámbito:

PCA (Análisis de los Componentes Principales):

Esta es una herramienta lineal que transforma variables de origen en otro conjunto totalmente nuevo de variables ortogonales, que se llaman principales componentes. Éstas, se ordenan de manera que la primera captura la mayor varianza posible de los datos, y cada componente subsiguiente captura la mayor varianza posible restante, bajo la restricción de ser ortogonal a las anteriores. Este análisis es bastante poderoso debido a su sencillez y a su notable eficacia en la

reducción de dimensionalidad cuando las relaciones en los datos son lineales *(Abdi & Williams, 2010)*.

t-Distributed Stochastic Neighbor Embedding (t-SNE)

El t-SNE es una técnica no lineal diseñada para la visualización de datos de alta dimensión en espacios de dos o tres dimensiones. Funciona modelando las similitudes entre pares de puntos de datos en el espacio original y busca preservar estas similitudes en el espacio reducido. Es particularmente eficaz para capturar estructuras locales complejas y es ampliamente utilizado para visualizar agrupaciones en los datos *(van der Maaten & Hinton, 2008)*.

Uniform Manifold Approximation and Projection (UMAP)

UMAP es una técnica de reducción de dimensionalidad que se basa en teorías de topología y geometría. Al igual que t-SNE, UMAP es una técnica no lineal que busca preservar tanto la estructura local como la global de los datos. Es conocida por su eficiencia computacional y su capacidad para mantener la integridad de la estructura de los datos en el espacio reducido, lo que la hace adecuada para aplicaciones de visualización y preprocesamiento de datos para aprendizaje automático *(McInnes, Healy, & Melville, 2018)*.

En resumen, la selección del algoritmo de reducción de dimensionalidad adecuado depende de la naturaleza de los datos y del objetivo específico del análisis. Mientras que PCA y es más apropiado para datos con relaciones lineales, técnicas como t-SNE y UMAP son preferibles cuando se busca capturar estructuras no lineales complejas en los datos.

3.1.2 Algoritmos de clasificación

Los algoritmos de clasificación son fundamentales en el aprendizaje automático supervisado, ya que permiten asignar etiquetas o categorías a nuevas observaciones basándose en patrones aprendidos de datos previamente

etiquetados. A continuación, se describen tres algoritmos de clasificación ampliamente utilizados:

Regresión Logística: La regresión logística es un método estadístico empleado para modelar la probabilidad de que una observación pertenezca a una de dos clases posibles. Aunque su nombre sugiere una relación con la regresión, se utiliza principalmente para tareas de clasificación binaria. Este algoritmo estima la probabilidad de ocurrencia de un evento al ajustar los datos a una función logística, produciendo valores entre 0 y 1. Es especialmente útil cuando se busca interpretar la influencia de variables independientes en la probabilidad de un resultado específico.

Random Forest: Random Forest es un algoritmo de ensamblado que construye múltiples árboles de decisión durante el entrenamiento y genera su resultado mediante la agregación de las predicciones de cada árbol individual, ya sea por votación mayoritaria en clasificación o promediando en regresión. Esta técnica mejora la precisión y controla el sobreajuste al reducir la varianza de los modelos individuales. Es robusta frente a datos faltantes y puede manejar grandes conjuntos de datos con numerosas características.

Gradient Boosting: Gradient Boosting es otro método de ensamblado que construye modelos de forma secuencial, donde cada modelo intenta corregir los errores del anterior. A diferencia de Random Forest, que construye árboles de decisión de manera independiente, Gradient Boosting ajusta nuevos modelos a los residuos de los modelos anteriores, mejorando iterativamente la precisión. Este enfoque es eficaz para manejar datos complejos y ha demostrado un rendimiento sobresaliente en diversas competencias de aprendizaje automático.

Entrenamiento y validación de modelos:

Para la predicción de fallos en los Access Points, se han implementado y ajustado tres modelos de clasificación: Regresión Logística, Random Forest y Gradient Boosting. A continuación, se detalla el proceso de entrenamiento y validación, incluyendo la preparación de los datos, la implementación de los modelos y la evaluación de su desempeño.

Primero, se realizó la selección y preparación de los datos. Estos fueron preprocesados eliminando atributos irrelevantes y transformando la variable objetivo en una clasificación binaria con los siguientes valores:

- 0: No hay evento.
- 1: Ocurre un evento (fallo en el Access Point).

La división de los datos se llevó a cabo utilizando un 70% para entrenamiento y un 30% para prueba, conservando la distribución original de las clases mediante estratificación. Posteriormente, se implementaron los modelos de clasificación. En la Tabla 4 se presentan los parámetros específicos utilizados en la configuración de cada uno.

Modelo	**Parámetro**	**Valor**
Regresión Logística	max_iter	1000
	class_weight	balanced
	random_state	42
Random Forest	n_estimators	100
	class_weight	balanced
	random_state	42
Gradient Boosting	n_estimators	100
	random_state	42
	sample_weight	Calculado según desbalance de clases

Tabla 4. Parámetros utilizados en los modelos de clasificación

La Regresión Logística, un modelo lineal que estima la probabilidad de pertenencia a una clase mediante una función sigmoide, fue implementada como

modelo base. Random Forest, un conjunto de árboles de decisión entrenados de manera independiente, fue utilizado para mejorar la precisión y reducir el sobreajuste. Finalmente, Gradient Boosting fue empleado por su capacidad para construir modelos secuenciales que corrigen los errores del conjunto anterior, lo cual resulta útil en contextos con ruido o patrones complejos.

3.2 Análisis de los resultados

Resultados

Para evaluar el desempeño de cada modelo, se utilizaron las siguientes métricas; **Exactitud**: Porcentaje de predicciones correctas, **Matriz de Confusión**: Comparación entre valores reales y predichos. **Reporte de Clasificación**: Incluye precisión, recall y F1-score. Los resultados obtenidos permiten comparar la capacidad predictiva de cada modelo y seleccionar el más adecuado para la predicción de fallos en Access Points, esto se puede ver en la siguiente tabla:

Modelo	Exactitud por Fold	Exactitud Promedio
Regresión Logística	0.5209, 0.7936, 0.7641, 0.7862, 0.7666	0.7263
Random Forest	1.0000, 1.0000, 1.0000, 1.0000, 1.0000	1.0000
Gradient Boosting	1.0000, 1.0000, 0.9975, 1.0000, 1.0000	0.9995

Tabla 5: Evaluación de los modelos

A pesar de que en los códigos de los anexos 2, 3 y 4; se implementan algunas formas de evitar el desbalance en las clases y proporcionar una evaluación más completa del desempeño de los modelos, se amplió el conjunto de métricas evaluadas mediante una validación cruzada estratificada con k=10. En lugar de reportar únicamente la exactitud, se calcularon también la precisión, el recall y el F1-score, métricas especialmente relevantes en contextos donde una clase es significativamente menos frecuente que la otra, como ocurre en el presente estudio.

Los resultados obtenidos se presentan en la Tabla 2. Puede observarse que los modelos de Random Forest y Gradient Boosting alcanzan valores de exactitud, precisión, recall y F1-score cercanos al 100%, lo cual resulta inusualmente alto. Este comportamiento sugiere un posible sobreajuste del modelo a los datos de entrenamiento, dado que se obtuvo un desempeño casi perfecto en todos los folds de validación, lo cual no es común en escenarios reales con ruido e incertidumbre.

En contraste, el modelo de Regresión Logística presentó un desempeño significativamente menor, con una precisión promedio de 0.0912 y un F1-score de 0.1148. Este bajo rendimiento sugiere que este modelo no logró capturar de forma efectiva las características discriminantes entre eventos y no eventos en el conjunto de datos.

Dado el riesgo de sobreajuste detectado, se propone como trabajo futuro realizar una validación con datos externos, es decir, un conjunto de datos no utilizado durante el entrenamiento ni la validación cruzada. Esto permitirá estimar con mayor realismo la capacidad de generalización de los modelos propuestos. Asimismo, puede considerarse la implementación de técnicas de regularización más estrictas o la reducción del número de árboles en los modelos de ensamble para mitigar el sobreajuste.

Modelo	**Exactitud**	**Precisión**	**Recall**	F1-score
Regresión Logística	6.493	912	4.214	1.148
Random Forest	9.997	9.875	1.2000	9.933
Gradient Boosting	9.997	9.875	1.2000	9.933

Tabla 6. Desempeño promedio de los modelos (validación cruzada k=10)

Análisis

El desempeño de los modelos evaluados mostró diferencias significativas en su capacidad para predecir eventos de fallo en Access Points, lo que permitió identificar las metodologías más efectivas para abordar este problema.

La Regresión Logística presentó una exactitud promedio del 72.63 %, con una alta variabilidad entre los diferentes folds de la validación cruzada. Esto indica que, aunque el modelo logra capturar algunos patrones en los datos, su capacidad para generalizar a nuevos ejemplos es limitada. La linealidad del modelo y la posible presencia de relaciones no lineales entre las métricas y los eventos podrían explicar su desempeño inferior en comparación con los modelos basados en árboles.

El modelo de Random Forest mostró un desempeño sobresaliente, alcanzando una exactitud del 100 % en todos los folds de la validación cruzada. Esto sugiere que el algoritmo es altamente efectivo para identificar los eventos de fallo en función de las métricas observadas. Sin embargo, es importante considerar la posibilidad de sobreajuste, ya que un rendimiento perfecto en el conjunto de entrenamiento no garantiza la misma capacidad de generalización en datos completamente nuevos.

Por su parte, Gradient Boosting también obtuvo resultados cercanos al 100 %, con una exactitud promedio de 99.95 %. Aunque no alcanzó la perfección absoluta de Random Forest, su desempeño estable y su capacidad de aprendizaje secuencial le otorgan una ventaja en términos de ajuste progresivo a los datos. A diferencia de Random Forest, que construye los árboles de manera independiente, Gradient Boosting optimiza cada nuevo árbol en función de los errores cometidos por los anteriores, lo que puede mejorar su capacidad de generalización.

La comparación de los resultados sugiere que los modelos basados en árboles son los más adecuados para abordar la predicción de fallos en redes Wi-Fi. No obstante, la elección final del modelo debe considerar no solo la exactitud, sino también otros factores como el tiempo de entrenamiento, la interpretabilidad y la robustez ante nuevas observaciones. En este caso, el alto desempeño de Random Forest y Gradient Boosting indica que ambos son opciones viables para su implementación en un sistema de monitoreo automatizado de redes, con una ligera ventaja para Gradient Boosting en términos de control sobre el ajuste del modelo.

Dado que los modelos obtuvieron una clasificación prácticamente perfecta en los datos evaluados, sería recomendable realizar pruebas adicionales con conjuntos de datos externos o incluir estrategias de validación más estrictas para confirmar su capacidad de generalización. Asimismo, podría explorarse la combinación de técnicas de reducción de dimensionalidad con la clasificación, evaluando si una representación más compacta de los datos mejora la interpretabilidad y estabilidad de los modelos sin comprometer la precisión en la detección de fallos.

3.3 Modelos de árboles de decisión

Se utilizaron árboles de decisión de baja profundidad para favorecer la interpretabilidad del modelo, lo cual es especialmente valioso en el contexto de una tesis enfocada en el análisis de métricas para la predicción de eventos. Limitar la profundidad a tres niveles permitió representar gráficamente el proceso de toma de decisiones del modelo de forma clara y comprensible, facilitando la identificación de umbrales críticos y la influencia relativa de cada variable. Esta aproximación no solo hace más transparente el comportamiento del modelo, sino que también permite a los investigadores y responsables técnicos comprender y validar fácilmente los patrones detectados por el sistema de clasificación.

Tras observar que los modelos basados en árboles de decisión obtenían resultados casi perfectos en la clasificación, se realizó un análisis detallado para identificar las variables que más influían en la predicción de eventos en los Access Points. Para ello, se entrenó un Árbol de Decisión con una profundidad máxima de tres niveles, lo que permitió visualizar de manera clara qué atributos separaban mejor las clases.

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt

# Definir y entrenar un árbol de decisión pequeño
tree_clf = DecisionTreeClassifier(max_depth=3, random_state=42)
tree_clf.fit(X_train, y_train)

# Graficar el árbol
plt.figure(figsize=(20, 10))
```

```
plot_tree(tree_clf, feature_names=X.columns, class_names=["No event",
"Event"], filled=True)
plt.show()
```

En la Figura 6 se observa que la variable `(Metric) Process count` fue la primera métrica seleccionada por el modelo como criterio de división. Con un umbral de 11.003, el modelo logra una separación perfecta entre ambas clases: todos los registros con un valor menor o igual al umbral corresponden a la clase "No event", mientras que los que lo superan pertenecen íntegramente a la clase "Event". El valor del índice de Gini en los nodos hijos es cero, lo que indica pureza total en ambas particiones. Este resultado evidencia que (Metric) Process count posee una capacidad discriminativa excepcional, al punto de dominar completamente la predicción, lo que motivó su posterior eliminación para analizar el comportamiento del modelo sin su influencia directa.

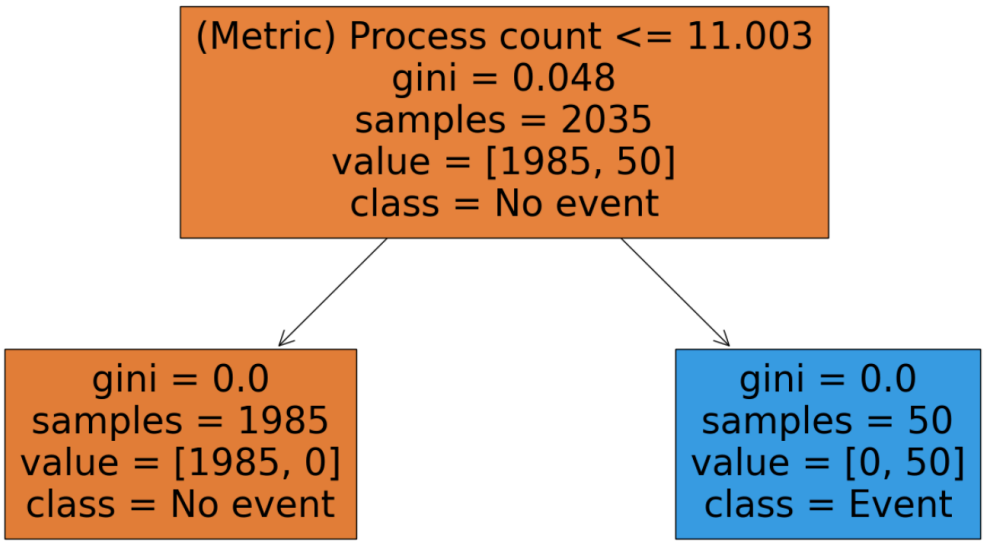

Figura 6. Árbol de decisión destacando `(Metric) Process count` como variable clave en la predicción de eventos. Fuente: Elaboración propia

Dado que `(Metric) Process count` parecía dominar la predicción, se eliminó del conjunto de datos para analizar el comportamiento del modelo sin su influencia. En la Figura 7 se observa el árbol resultante tras eliminar la métrica (Metric) Process count, previamente identificada como la principal variable predictiva. En su ausencia, el modelo seleccionó (Metric) Process CPU como nuevo criterio de división principal, dividiendo perfectamente los datos con un umbral de 0.011. Esta nueva variable permite separar con pureza total los casos entre las clases “No event” y “Event”, con un índice de Gini igual a cero en ambos nodos hijos. Este comportamiento revela que, aunque algunas métricas dominan la predicción inicial, otras también contienen suficiente información discriminativa para realizar una clasificación precisa. La eliminación progresiva de variables permitió descubrir jerarquías de importancia entre las métricas y entender mejor cómo cada una contribuye a la detección de eventos en los Access Points.

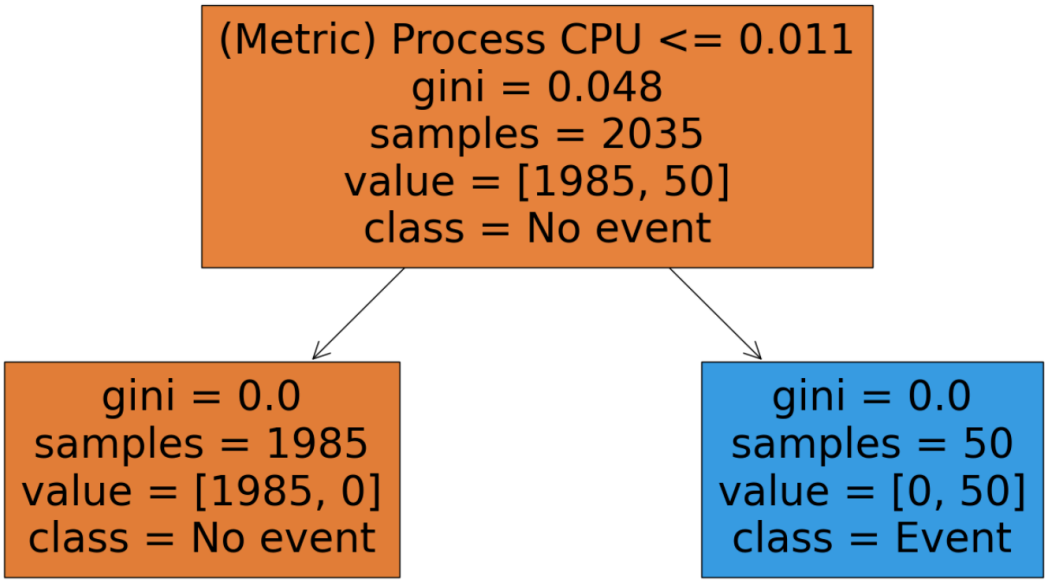

Figura 7. Árbol de decisión tras eliminar `(Metric) Process count`, ahora liderado por `(Metric) Process CPU`. Fuente: Elaboración propia

Siguiendo el mismo procedimiento, `(Metric) Process CPU` fue removida del dataset y se volvió a entrenar el modelo. En la Figura 8 se presenta el árbol de decisión resultante tras eliminar la métrica (Metric) Process CPU, lo que permitió que (Metric) System CPU usage se posicionara como el nuevo nodo raíz. Esta métrica, con un umbral de 1.701, separa correctamente la mayoría de los casos de la clase "No event", mientras que el grupo restante es evaluado por la variable (Metric) Process memory. Esta última introduce una segunda división que permite aislar con precisión los casos de "Event", con sólo una muestra mal clasificada. El árbol conserva una estructura de baja profundidad y alta pureza, lo que evidencia que, incluso sin las métricas dominantes eliminadas previamente, el modelo conserva una notable capacidad de discriminación. Este resultado reafirma la utilidad de estas variables en la predicción de eventos y refuerza la estrategia de eliminación progresiva como herramienta para revelar la relevancia estructural de distintas métricas en la clasificación.

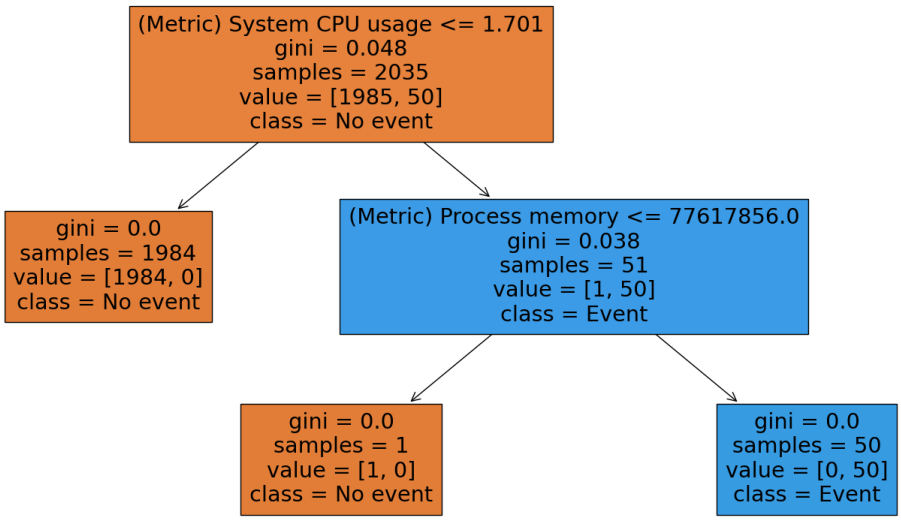

Figura 8. Árbol de decisión sin `(Metric) Process CPU`, resaltando `(Metric) System CPU usage` y `(Metric) Process memory`. Fuente: Elaboración propia

Esta parte del trabajo permitió identificar estas métricas como un factor clave en la predicción de eventos en los Access Points. La eliminación progresiva de variables dominantes ayudó a comprender mejor el peso de cada atributo en la clasificación y a determinar cuáles métricas tienen un impacto significativo en la ocurrencia de fallos.

La selección de los algoritmos Regresión Logística, Random Forest y Gradient Boosting se basó en criterios de interpretabilidad, rendimiento y adecuación al tipo de datos disponibles. En primer lugar, la Regresión Logística fue elegida por su simplicidad y facilidad de interpretación, lo que resulta útil para establecer relaciones lineales entre las métricas de los Access Points y la probabilidad de ocurrencia de un evento. Esta característica la convierte en un buen punto de partida para evaluar el poder predictivo de las variables. Por otro lado, Random Forest y Gradient Boosting fueron seleccionados como representantes de modelos de ensamble, capaces de capturar relaciones no lineales y patrones complejos en los datos, ofreciendo generalmente un mejor rendimiento predictivo que los modelos lineales simples. Ambos algoritmos han demostrado ser robustos ante el sobreajuste y eficaces en tareas de clasificación con conjuntos de datos similares.

En contraste con otras alternativas como SVM o redes neuronales, las opciones seleccionadas ofrecen ventajas significativas en términos de interpretación y eficiencia computacional. Si bien los SVM pueden ofrecer buen rendimiento en contextos de alta dimensionalidad, su funcionamiento y resultados suelen ser menos transparentes, y su ajuste requiere una cuidadosa selección de kernels y parámetros. Las redes neuronales, por su parte, tienden a requerir mayores volúmenes de datos, mayor poder de cómputo y presentan una menor interpretabilidad, lo cual las hace menos adecuadas para un problema donde la explicación de los resultados y la comprensión de las variables influyentes es tan importante como la precisión del modelo. En este contexto, los algoritmos seleccionados representan un equilibrio entre rendimiento, transparencia y facilidad de implementación, ajustándose adecuadamente a los objetivos del análisis.

Conclusiones y recomendaciones

Conclusiones y recomendaciones

Esta investigación desarrolló modelos de clasificación para la predicción de fallos en Access Points (APs) de redes Wi-Fi, a partir de una base de datos construida mediante la integración de métricas temporales y eventos críticos. El preprocesamiento incluyó imputación de valores faltantes, normalización y reducción de dimensionalidad con UMAP, lo que permitió optimizar el conjunto de datos para su análisis predictivo.

Se entrenaron y evaluaron modelos de Regresión Logística, Random Forest y Gradient Boosting, incorporando técnicas más robustas de balanceo de clases y validación cruzada. Los modelos basados en árboles de decisión, especialmente Random Forest, alcanzaron niveles de exactitud cercanos al 100 %, lo que motivó un análisis de importancia de variables. Se identificó que las métricas (Metric) Process Count, (Metric) Process CPU y (Metric) System CPU usage son altamente predictivas de la ocurrencia de eventos, lo que indica una fuerte relación entre el uso de recursos del sistema y la probabilidad de fallo en los APs.

Estos hallazgos responden al objetivo general del estudio: generar las condiciones para una gestión proactiva de redes Wi-Fi mediante la predicción anticipada de fallos. Como recomendación práctica, se sugiere la integración de los modelos desarrollados en sistemas de monitoreo en tiempo real, a fin de emitir alertas preventivas y reducir interrupciones del servicio. Para investigaciones futuras, se propone validar el desempeño de los modelos con datos de diferentes entornos operativos y explorar arquitecturas basadas en redes neuronales profundas que permitan mejorar la capacidad de generalización del sistema predictivo a voluntad.

Fuentes de consulta

- Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433–459. https://doi.org/10.1002/wics.101
- Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. SIAM Review, 37(4), 573–595. https://doi.org/10.1137/1037127
- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
- Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215–232. https://doi.org/10.1111/j.2517-6161.1958.tb00292.x
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451
- Gill, P., Jain, N., & Nagappan, N. (2011, August). Understanding network failures in data centers: Measurement, analysis, and implications. In Proceedings of the ACM SIGCOMM 2011 Conference (pp. 350–361). ACM. https://doi.org/10.1145/2018436.2018477
- Lam, H. S., Tan, Y. F., Soo, W. K., Guo, X., & Lee, Z. M. (2016). Network equipment failure prediction with big data analytics. International Journal on Soft Computing Applications, 8(3), 1–15.
- McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426. https://arxiv.org/abs/1802.03426
- Myrzatay, A., Rzayeva, L., Bandini, S., Shaye, I., Saoud, B., Çolak, I., & Kayisli, K. (2021). Predicting LAN switch failures: An integrated approach with DES and machine learning techniques (RF/LR/DT/SVM). IEEE Transactions on Network and Service Management, 18(2), 100–110. https://doi.org/10.1016/j.rineng.2024.102356
- van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.
- Wang, Z., Zhang, M., Wang, D., Song, C., Liu, M., Li, J., Lou, L., & Liu, Z. (2019). Failure prediction using machine learning and time series in optical network. Optics Express, 27(3), 10–12. https://doi.org/10.1364/OE.25.018553
- Averineni, A. (2025). *Leveraging machine learning for anomaly detection in telecom network management*. *Journal of Computer Science and Technology Studies*, *7*(4), 1–10. https://doi.org/10.32996/jcsts.2025.7.4.2
- Hasan, K., Trappenberg, T., & Haque, I. (2024). *A generalized transformer-based radio link failure prediction framework in 5G RANs*. arXiv. https://doi.org/10.48550/arXiv.2407.05197
- Nabeel, M., Nimara, D. D., & Zanouda, T. (2024). *Fault detection in mobile networks using diffusion models*. arXiv. https://doi.org/10.48550/arXiv.2404.09240
- Bourgerie, R., & Zanouda, T. (2023). *Fault detection in telecom networks using bi-level federated graph neural networks*. arXiv. https://doi.org/10.48550/arXiv.2311.14469

ANEXOS

ANEXO 1: creator-dataframe.py

Este código en Python carga dos archivos CSV (uno con eventos y otro con métricas de Access Points), filtra los datos en torno a una fecha específica, pivotea las métricas y los eventos, y los combina en un solo DataFrame que se exporta como un nuevo archivo CSV. Durante el proceso, se realiza la imputación de valores nulos, se renombran etiquetas para distinguir entre métricas y eventos, y se genera una variable binaria que indica la presencia del evento “Access points down threshold exceeded”. Todo esto tiene como fin construir una base de datos lista para análisis de correlación o clasificación.

```
import pandas as pd

FOLDER_PATH = "Data/"
E_FILE_NAME = "events2024"
M_FILE_NAME = "metrics202403"
FECHA = "2024-03-02 18:00:00"
PERIODO = "30"

def procesar_df(dataframe_e, dataframe_m, path, fecha_fija, period, specific_event = None,
event_historico = None):
    """Esta función procesa los dataframes, los filtra, pivotea y une mediante un merge"""
    # Construcción de la ruta completa a los archivos
    e_file_path = str(path)+str(dataframe_e) + ".csv"
    m_file_path = str(path)+str(dataframe_m) + ".csv"

    # Carga de datos
    metrics = pd.read_csv(m_file_path)
    events = pd.read_csv(e_file_path)

    # Procesamiento de strings
    metrics['metric'] = metrics['metric'].apply(lambda x: "(Metric) " + x)
    events['classification'] = events['classification'].apply(lambda x: "(Event) " + x)
    events["firstSeen"] = pd.to_datetime(events["firstSeen"], format='ISO8601').dt.floor('s')
    metrics['timestamp'] = pd.to_datetime(metrics['timestamp'])

    if event_historico is not None:
        #events = events[events["classification"] == "(Event) " + event_historico]
        events, date_lim, metrics = filtrar_tiempo(date = fecha_fija, data = events,
data_metrics=metrics, minutos= period, is_historical=event_historico)
        name_for_file = "Historico_MatrizCorrelacion_" + date_lim.replace(":", "-").replace(" ", "-")
    else:
        # Filtrar para el tiempo
        events, date_lim, metrics= filtrar_tiempo(date = fecha_fija, data = events, minutos= period,
data_metrics= metrics)
        #metrics = metrics[metrics["timestamp"] == fecha_fija]
        partes = str(date_lim).split(" ")
        date_lim = partes[1]
        name_for_file = "MatrizCorrelacion_" + fecha_fija.replace(" ", "_").replace(":", "-") + "_" +
date_lim.replace(":", "-")

    # Agrupar y eliminar columna de tiempo
    df_events_grouped = events.drop("firstSeen", axis = 1)
    df_metrics_grouped = metrics.drop("timestamp", axis = 1)
```

```
    df_events_grouped = df_events_grouped.groupby(["device_id", "classification"]).sum().reset_index()
    df_metrics_grouped = df_metrics_grouped.groupby(["device_id", "metric"]).mean().reset_index()

    # Filtrado y pivotado de datos
    events_filtered = df_events_grouped[df_events_grouped["classification"] != "(Event) Monitoring
error"]
    if specific_event is not None:
        evento_string = "(Event) " + specific_event
        events_filtered = df_events_grouped[df_events_grouped["classification"] == evento_string]
        specific_event = specific_event.replace(" ", "-")
    pivoted_events = events_filtered.pivot_table(index="device_id", columns="classification",
values="Total").reset_index()
    pivoted_metrics = df_metrics_grouped.pivot_table(index='device_id', columns='metric',
values='average').reset_index()

    # Merge de los dataframes
    data_merge = pd.merge(pivoted_events, pivoted_metrics, on='device_id', how='outer') #how = "inner"
    data_merge['(Event) Access points down threshold exceeded'] = data_merge['(Event) Access points down
threshold exceeded'].apply(
    lambda x: 1 if isinstance(x, (int, float)) and x > 0 else 0)

    return data_merge, name_for_file, date_lim, specific_event
def filtrar_tiempo(date, data, data_metrics, minutos = "60", is_historical = None):
    """Esta función filtra el dataframe según lo que se requiera, 30 o 60 minutos"""
    intervalo = pd.Timestamp(date) + pd.Timedelta(minutes=int(minutos))
    df = data[(data["firstSeen"] > pd.Timestamp(date) + pd.Timedelta(minutes=1)) & (data["firstSeen"] <=
intervalo)]
    metrics = data_metrics[data_metrics["timestamp"] == date]
    if is_historical is not None:
        data["rounded_hour"] = data["firstSeen"].dt.floor("60T")
        df = data[(data["firstSeen"] >= data["rounded_hour"] + pd.Timedelta(minutes=1)) &
(data["firstSeen"] <= data["rounded_hour"] + pd.Timedelta(minutes=int(minutos)))]
        metrics = data_metrics[data_metrics["timestamp"].isin(df["rounded_hour"])]# ==
df["rounded_hour"]]
        max_time = metrics["timestamp"].dt.date.max()
        min_time = metrics["timestamp"].dt.date.min()
        intervalo = "Eventos_" + str(min_time) + "_" + str(max_time) + "_Periodo " + PERIODO + "
minutos"
        df = df.drop("rounded_hour", axis = 1)
    return df, intervalo, metrics


df_merged, files_name, end_date, evento_tag  = procesar_df(E_FILE_NAME, M_FILE_NAME, FOLDER_PATH, FECHA,
PERIODO, event_historico="Access points down threshold exceeded", specific_event="Access points down
threshold exceeded")#
df_merged = df_merged.apply(lambda col: col.fillna(col.mean()) if col.dtype.kind in 'fi' else col)
df_merged.to_csv('data.csv', index=False)
```

El resultado de ejecutar este código es un archivo CSV, cuyas columnas son el id del dispositivo, una clasificación (Evento o No evento), y el resto de sus métricas.Un ejemplo de alguna de sus filas es:

“AAAAA--Au24Mzmg4rrtg0Yc9bTg=,(Event) No event,0.0 ,98.7 ,0.8,0.0,0.1,52.5,1.3,0.4,2.4,85.4,59.7,20.5,0.0,98.7,0.3,0.0,2.4,11.0,11.0, 216.4,0.0,11.0,83399680.0,100.0,1.5,1.2,3131021955.3,5.0,725.6”

ANEXO 2: Implementación de regresión logística

Este código implementa un modelo de regresión logística para predecir eventos en Access Points. Se parte de un conjunto de datos procesado donde la variable objetivo es binaria: 0 representa ausencia de evento y 1 la presencia de cualquier tipo de fallo. El conjunto se divide en entrenamiento y prueba, respetando la proporción original de clases. El modelo se entrena usando class_weight='balanced' para mitigar el desbalance de clases. Posteriormente, se evalúa el rendimiento del modelo con métricas como exactitud, matriz de confusión y reporte de clasificación.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Cargar dataset
df = pd.read_csv('/home/hugo/INFOTEC/TesisMCDI/Data/dataSanosUnicos.csv')

# Eliminar columna no relevante para el modelo
X = df.drop(['device_id', 'classification'], axis=1)

# Convertir la variable objetivo a binaria: 0 = "No event", 1 = otros eventos
y = np.where(df['classification'] == '(Event) No event', 0, 1)

# Dividir en datos de entrenamiento y prueba (70% - 30%)
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.3,
    stratify=y,  # Mantener proporción de clases
    random_state=42,
    shuffle=True #Mezclamos los datos antes de dividirlos
)

# Verificación rápida
print("Distribución de clases en entrenamiento:", np.unique(y_train, return_counts=True))
print("Distribución de clases en prueba:", np.unique(y_test, return_counts=True))
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Entrenar modelo con class_weight balanceado
lr = LogisticRegression(max_iter=1000, random_state=42, class_weight="balanced")
lr.fit(X_train, y_train)

# Predecir y evaluar
y_pred = lr.predict(X_test)

print("=== Regresión Logística ===")
print(f"Exactitud: {accuracy_score(y_test, y_pred):.2f}")
print("\nMatriz de confusión:")
print(confusion_matrix(y_test, y_pred))
print("\nReporte de clasificación:")
print(classification_report(y_test, y_pred))
```

Este código produce un resultado como el siguiente:

```
=== Regresión Logística ===
Exactitud: 0.75

Matriz de confusión:
[[657 195]
 [ 21   0]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.97      0.77      0.86       852
           1       0.00      0.00      0.00        21

    accuracy                           0.75       873
   macro avg       0.48      0.39      0.43       873
weighted avg       0.95      0.75      0.84       873
```

ANEXO 3: Implementación de Random Forest

Este código emplea un clasificador Random Forest para predecir la ocurrencia de eventos en dispositivos de red. Se utiliza el mismo conjunto de entrenamiento y prueba generado en el anexo anterior. El modelo se ajusta con 100 árboles (n_estimators=100) y se aplica balanceo de clases mediante class_weight='balanced'. La evaluación del modelo incluye la exactitud global, matriz de confusión y métricas de precisión, recall y F1-score, lo que permite analizar su desempeño en escenarios con clases desbalanceadas.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Entrenar modelo con class_weight balanceado
rf = RandomForestClassifier(n_estimators=100, random_state=42, class_weight="balanced")
rf.fit(X_train, y_train)

# Predecir y evaluar
y_pred = rf.predict(X_test)

print("=== Random Forest ===")
print(f"Exactitud: {accuracy_score(y_test, y_pred):.2f}")
print("\nMatriz de confusión:")
print(confusion_matrix(y_test, y_pred))
print("\nReporte de clasificación:")
print(classification_report(y_test, y_pred))
```

Este código produce un resultado como el siguiente:

```
=== Random Forest ===
Exactitud: 1.00

Matriz de confusión:
[[852   0]
 [  0  21]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       852
           1       1.00      1.00      1.00        21

    accuracy                           1.00       873
   macro avg       1.00      1.00      1.00       873
weighted avg       1.00      1.00      1.00       873
```

ANEXO 4: Implementación de Gradient Boosting

Este código aplica el algoritmo de Gradient Boosting para clasificar dispositivos según la presencia o ausencia de fallos. A diferencia de los modelos anteriores, se emplean pesos individuales para cada muestra utilizando compute_sample_weight con el objetivo de compensar el desbalance de clases. El modelo se entrena con 100 iteraciones y se evalúa usando las mismas métricas estándar. Esto permite comparar su rendimiento frente a los modelos anteriores en condiciones equivalentes.

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.utils.class_weight import compute_sample_weight
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Calcular pesos para cada muestra según la distribución de clases
sample_weights = compute_sample_weight(class_weight="balanced", y=y_train)

# Entrenar modelo con sample_weight
gb = GradientBoostingClassifier(n_estimators=100, random_state=42)
gb.fit(X_train, y_train, sample_weight=sample_weights)

# Predecir y evaluar
y_pred = gb.predict(X_test)

print("=== Gradient Boosting ===")
print(f"Exactitud: {accuracy_score(y_test, y_pred):.2f}")
print("\nMatriz de confusión:")
print(confusion_matrix(y_test, y_pred))
print("\nReporte de clasificación:")
print(classification_report(y_test, y_pred))
```

Este código produce un resultado como el siguiente:

```
=== Gradient Boosting ===
Exactitud: 1.00

Matriz de confusión:
[[852   0]
 [  0  21]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       852
           1       1.00      1.00      1.00        21

    accuracy                           1.00       873
   macro avg       1.00      1.00      1.00       873
weighted avg       1.00      1.00      1.00       873
```

Estos códigos se muestran en texto plano en la siguiente URL:

https://github.com/HugoAceves/ANEXOS-Predicci-n-de-Fallos-en-redes-Wi-Fi-mediante-aprendizaje-computacional

Índice de términos